

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-148584

(43)Date of publication of application : 30.05.2000

(51)Int.Cl.

G06F 12/08

G06F 9/38

(21)Application number : 10-318759

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing : 10.11.1998

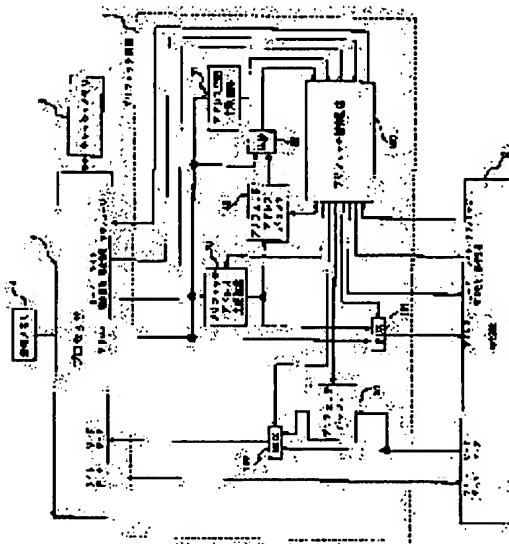
(72)Inventor : TANAKA TETSUYA

## (54) PREFETCHING METHOD AND DEVICE

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To shorten the waiting time of a processor and to make it possible to execute a high speed program without requiring the generation of a prefetching instruction by a compiler and changing data stored in a cache memory.

**SOLUTION:** In the prefetching method, a state that a read request address from a processor 4 to a main storage 2 is a prefetching space is discriminated, and then data are prefetched and stored in a prefetch buffer 30 independently of the operation of the processor 4. When the processor 4 requests data stored in the buffer 30, data are supplied from the buffer 30 to the processor 4.



## LEGAL STATUS

[Date of request for examination]

05.10.2004

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]



**【特許請求の範囲】**

**【請求項1】** プロセッサの記憶デバイスへのリード要求アドレスがプリフェッチ空間であることを判別する空間判別手段と、プリフェッチアドレスを生成するアドレス生成手段と、このアドレス生成手段で得た前記プリフェッチアドレスを用いて前記記憶デバイスから対応するデータを読み出すデータ読み出し手段と、前記プロセッサの前記記憶デバイスへのリード要求アドレスが前記空間判別手段により前記プリフェッチ空間であることが判別された場合は前記プロセッサの前記記憶デバイスへのリード要求とは無関係に前記データ読み出し手段によりデータを読み出して保持するプリフェッチバッファを有するプリフェッチ手段と、前記プロセッサの前記記憶デバイスへのリード要求アドレスが前記プリフェッチバッファに保持されているデータのアドレスと一致することを検出するアドレス一致検出手段と、このアドレス一致検出手段により一致が検出された場合は、前記プリフェッチバッファに保持されているデータを前記プロセッサに転送するデータ転送手段とを備えたプリフェッチ装置。

**【請求項2】** プリフェッチ手段におけるデータの読み出しを繰り返す反復プリフェッチ手段と、この反復プリフェッチ手段を終了させるプリフェッチ終了手段とを備えた請求項1記載のプリフェッチ装置。

**【請求項3】** 反復プリフェッチ手段の行われた回数を計数する回数計数手段と、制御レジスタに設定された値と前記回数計数手段により計数されたプリフェッチ手段の回数とが一致することを検出する回数一致手段と、この回数一致手段により一致を検出した場合は前記反復プリフェッチ手段を終了する手段を備えた請求項2記載のプリフェッチ装置。

**【請求項4】** 反復プリフェッチ手段によりプリフェッチバッファに保持したデータの大きさを計数するデータサイズ計数手段と、制御レジスタに設定されたデータサイズと前記データサイズ計数手段により得られた前記プリフェッチバッファに保持されているデータの大きさとが一致することを検出するデータサイズ一致検出手段と、このデータサイズ一致検出手段により一致が検出された場合は前記反復プリフェッチ手段を終了する手段を備えた請求項2記載のプリフェッチ装置。

**【請求項5】** 反復プリフェッチ手段のアドレス生成手段により生成したアドレスが制御レジスタに設定されたアドレスと一致することを検出する終了アドレス検出手段と、この終了アドレス検出手段により一致が検出された場合は、前記反復プリフェッチ手段を終了する手段を備えた請求項2記載のプリフェッチ装置。

**【請求項6】** アドレス生成手段が、プロセッサの記憶デバイスへのリード要求アドレスに制御レジスタに設定された値を加算することでアドレスを生成する手段である請求項1記載のプリフェッチ装置。

**【請求項7】** アドレス生成手段が、プロセッサの記憶デバイスへのリード要求アドレスに対し、制御レジスタに設定された情報により決定される演算を施すことでアドレスを生成する手段である請求項1記載のプリフェッチ装置。

**【請求項8】** プロセッサにより制御レジスタを書き換える手段を備えた請求項3から請求項7のいずれかの1項に記載のプリフェッチ装置。

**【請求項9】** プリフェッチ手段におけるデータの読み出しを繰り返す第1と第2の反復プリフェッチ手段と、プロセッサの記憶デバイスへの第1のリード要求アドレスにより第1の反復プリフェッチ手段が開始され、前記プロセッサの前記記憶デバイスへの第2のリード要求アドレスと第1の反復プリフェッチ手段によりプリフェッチバッファに保持されているデータのアドレスとが一致していないことがアドレス一致検出手段により検出された場合は、第1の反復プリフェッチ手段を終了し、第2の反復プリフェッチ手段を開始する手段を備えた請求項1記載のプリフェッチ装置。

**【請求項10】** データ転送手段によりプリフェッチバッファのデータをプロセッサに転送した場合は、プリフェッチバッファに保持されているデータをプリフェッチバッファから無効化する無効化手段を備えた請求項1記載のプリフェッチ装置。

**【請求項11】** アドレス一致検出手段によりプロセッサの記憶デバイスへのリード要求アドレスとプリフェッチバッファに保持したデータのアドレスとが一致しないことを検出した場合は、プリフェッチバッファに保持されたデータを無効化する手段を備えた請求項1記載のプリフェッチ装置。

**【請求項12】** プロセッサの記憶デバイスへのリード要求アドレスがプリフェッチ空間であることを判別する空間判別ステップと、プリフェッチアドレスを生成するアドレス生成ステップと、このアドレス生成ステップで得た前記プリフェッチアドレスを用いて前記記憶デバイスから対応するデータを読み出すデータ読み出しステップと、前記プロセッサの前記記憶デバイスへのリード要求アドレスが前記空間判別ステップにより前記プリフェッチ空間であることが判別された場合は前記プロセッサの前記記憶デバイスへのリード要求とは無関係に前記データ読み出しステップによりデータを読み出し、プリフェッチバッファに保持するプリフェッチステップと、前記プロセッサの前記記憶デバイスへのリード要求アドレスが前記プリフェッチバッファに保持されているデータのアドレスと一致することを検出するアドレス一致検出ステップと、このアドレス一致検出ステップにより一致が検出された場合は、前記プリフェッチバッファに保持されているデータを前記プロセッサに転送するデータ転送ステップとを含むプリフェッチ方法。

**【請求項13】** プリフェッチステップにおけるデータ

の読み出しを繰り返す反復プリフェッチステップと、この反復プリフェッチステップを終了させるプリフェッチ終了ステップとを含む請求項12記載のプリフェッチ方法。

【請求項14】 反復プリフェッチステップの行われた回数を計数する回数計数ステップと、制御レジスタに設定された値と前記回数計数ステップにより計数された前記プリフェッチステップの回数とが一致することを確認する回数一致ステップと、前記回数一致ステップにより一致を確認した場合は前記反復プリフェッチステップを終了するステップを含む請求項13記載のプリフェッチ方法。

【請求項15】 反復プリフェッチステップによりプリフェッチバッファに保持したデータの大きさを計数するデータサイズ計数ステップと、制御レジスタに設定されたデータサイズと前記データサイズ計数ステップにより得られた前記プリフェッチバッファに保持されているデータの大きさとが一致することを確認するデータサイズ一致検出ステップと、このデータサイズ一致検出ステップにより一致が検出された場合は前記反復プリフェッチステップを終了するステップを含む請求項13記載のプリフェッチ方法。

【請求項16】 反復プリフェッチステップのアドレス生成ステップにより生成したアドレスが制御レジスタに設定されたアドレスと一致することを確認する終了アドレス検出ステップと、この終了アドレス検出ステップにより一致が検出された場合は、前記反復プリフェッチステップを終了するステップを含む請求項13記載のプリフェッチ方法。

【請求項17】 アドレス生成ステップが、プロセッサの記憶デバイスへのリード要求アドレスに制御レジスタに設定された値を加算することでアドレスを生成するステップである請求項12記載のプリフェッチ方法。

【請求項18】 アドレス生成ステップが、プロセッサの記憶デバイスへのリード要求アドレスに対し、制御レジスタに設定された情報により決定される演算を施すことでアドレスを生成するステップである請求項1記載のプリフェッチ方法。

【請求項19】 プロセッサにより制御レジスタを書き換えるステップを含む請求項14から請求項18のいずれかの1項に記載のプリフェッチ方法。

【請求項20】 プリフェッチステップにおけるデータの読み出しを繰り返す第1と第2の反復プリフェッチステップと、プロセッサの記憶デバイスへの第1のリード要求アドレスにより前記第1の反復プリフェッチステップが開始され、前記プロセッサの前記記憶デバイスへの第2のリード要求アドレスと前記第1の反復プリフェッチステップによりプリフェッチバッファに保持されているデータのアドレスとが一致していないことがアドレス一致検出ステップにより検出された場合は、前記第1の

反復プリフェッチステップを終了し、前記第2の反復プリフェッチステップを開始するステップを含む請求項12記載のプリフェッチ方法。

【請求項21】 データ転送ステップによりプリフェッチバッファのデータをプロセッサに転送した場合は、プリフェッチバッファに保持されている前記データを前記プリフェッチバッファから無効化する無効化ステップを含む請求項12記載のプリフェッチ方法。

【請求項22】 前記アドレス一致検出ステップによりプロセッサの記憶デバイスへのリード要求アドレスとプリフェッチバッファに保持したデータのアドレスとが一致しないことを確認した場合は、プリフェッチバッファに保持されたデータを無効化するステップを含む請求項12記載のプリフェッチ方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、プロセッサにおけるプリフェッチ方法および装置に関し、特に記憶デバイスからのデータフェッチの待ち時間を低減するためデータフェッチに先行してあらかじめデータフェッチをしておくプリフェッチ技術に関する。

【0002】

【従来の技術】近年、マイクロプロセッサ応用製品の高機能化および高速化に伴い、メディア処理に代表される信号処理に使われる場合が増えてきた。信号処理ではメモリ上のデータを読み出し、演算後、メモリに書き込むことをデータの集合に対して行うストリームフィルタ処理が数多く現れる。

【0003】ストリームフィルタ処理で扱うデータ集合の個々のデータは、マイクロプロセッサ（以下、単に「プロセッサ」という）で再利用することが少ないため、キャッシュがヒットすることは期待できない。これはストリームフィルタで取り扱うデータがプロセッサ外部で生成され、プロセッサで処理した後はプロセッサ外部で使用されることによる。

【0004】プロセッサはこのようなデータをロードするとき、メモリアクセスの待ち時間の分だけ処理が待たされることになる。この待ち時間を低減するために用いられる技術のひとつにプリフェッチがある。従来のプリフェッチ装置およびプリフェッチ方法としては、プリフェッチ命令で実現するものと、キャッシュメモリによりプリフェッチ機能を実現するものがある。

【0005】プリフェッチ命令は例えば、特開平8-55025号に開示されているように、通常のロード命令として機能した後、ロードしたデータを含むブロック（例えば16バイト）をメモリから読み出し、プリフェッチバッファに保持するものである。この時読み出したブロックのアドレスも同時に保持される。プリフェッチ命令実行後、他のロード命令が実行されメモリにリード要求を出したとき、要求されたアドレスがプリフェッチ

バッファに保持されたブロック内のデータを要求している場合はプリフェッチバッファから要求されたデータをプロセッサに供給する。メモリアクセスの待ち時間はプリフェッチ命令の時点から生じるが、プロセッサの動作とは無関係に動作するのでプロセッサに待ち時間が生じることはない。プリフェッチ命令のメモリアクセスが完了した後にロード命令でプリフェッチしたデータ（例えば(A+4)のデータ)を読み出す場合はプロセッサのメモリアクセス待ち時間を削減できる。

【0006】図10に従来のプリフェッチ命令によるプリフェッチ装置の動作説明図を示す。“ldpf(A), r0”命令はアドレス(A)のデータをメモリから読み出し(ロードし)レジスタr0に格納する命令であり、アドレス(A)に対するロードを実行後、アドレス(A+4)以降のデータを次々にメモリから読み出しプリフェッチバッファに保持する。“ld(A+4), r1”命令はアドレス(A+4)のデータをロードしレジスタr1に保持する命令である。“ldpf(A), r0”命令の実行後はアドレスAのデータをロードするため待ち時間が生じているが、“ld(A+4), r1”命令を実行するときはすでにプリフェッチバッファに所望のデータが保持されており、プリフェッチバッファからデータを供給するため、ロードの待ち時間は生じることがない。

【0007】キャッシュメモリによりプリフェッチ機能を実現する場合は、キャッシュメモリの各エントリがプリフェッチバッファの役目を果たす。最初のロード命令で、あるアドレスのデータをメモリから読み出す場合、キャッシュメモリの管理単位であるキャッシュブロックがキャッシュミスによって適切なエントリにフェッチされる。これによりプリフェッチと同様の効果を得ることができる。図11に従来のキャッシュメモリによるプリフェッチ装置の動作説明図を示す。“ld(A), r0”命令はアドレス(A)のデータをメモリから読み出しレジスタr0に保持する命令である。このロード命令がキャッシュミスを起こす場合、キャッシュメモリはアドレス(A)のデータを含むキャッシュブロックをメモリから読み出し、適切なエントリに保持する(リフィルする)。このキャッシュブロックにはアドレス(A+4)のデータも含まれているとすると、次のロード命令“ld(A+4), r1”が実行される場合、アドレス(A+4)のデータは既にキャッシュメモリの適切なエントリに格納されているので、メモリからの読み出しを行わずにキャッシュメモリからデータを転送することで、アドレス(A+4)に対するメモリ読み出しの待ち時間は生じない。

【0008】

【発明が解決しようとする課題】しかしながら、上記従来技術のプリフェッチ命令によるプリフェッチ装置ではコンパイラによって命令を生成することが困難であると

言う問題を有している。すなわち、プリフェッチ命令による実現は、プリフェッチ命令と通常のロード命令とを区別して使う必要があるが、一般的なC言語などの高級言語はストリームフィルタで使用されるデータ(以下、単に「ストリームデータ」と言う)であることを指定することができないため、プリフェッチ命令と通常のロード命令を区別して生成することが困難である。また、言語処理系でプリフェッチ命令の有効な場合を自動的に判別することは非常に広範囲にわたるプログラムの意味解析が必要になるため同様に困難である。

【0009】一方、上記従来技術のキャッシュメモリによるプリフェッチ装置では、ストリームデータをキャッシュのエントリに格納するため、そのエントリにあった他のデータがキャッシュメモリから追い出されてしまう。ストリームデータはキャッシュメモリに残っていても再利用されることはないが、キャッシュメモリから追い出されたデータは再利用する可能性のあるデータである。もし、追い出されたデータを再利用する場合は、キャッシュミスが生じ性能の低下を招く。すなわち、本来キャッシュメモリに格納しておいた方が良いデータをストリームデータで置き換えることで性能低下を招く可能性があるという問題を有している。

【0010】この発明は、上記の問題を解決するもので、コンパイラでプリフェッチ命令を生成する必要がなく、キャッシュメモリに保持されているデータが変化することなく、プロセッサの待ち時間を低減でき、高速のプログラムの実行が行なえるプリフェッチ方法および装置を提供することを目的としている。

【0011】

【課題を解決するための手段】請求項1記載のプリフェッチ装置は、プロセッサの記憶デバイスへのリード要求アドレスがプリフェッチ空間であることを判別する空間判別手段と、プリフェッチアドレスを生成するアドレス生成手段と、このアドレス生成手段で得たプリフェッチアドレスを用いて記憶デバイスから対応するデータを読み出すデータ読み出し手段と、プロセッサの記憶デバイスへのリード要求アドレスが空間判別手段によりプリフェッチ空間であることが判別された場合はプロセッサの記憶デバイスへのリード要求とは無関係にデータ読み出し手段によりデータを読み出して保持するプリフェッチバッファを有するプリフェッチ手段と、プロセッサの記憶デバイスへのリード要求アドレスがプリフェッチバッファに保持されているデータのアドレスと一致することを検出するアドレス一致検出手段と、このアドレス一致検出手段により一致が検出された場合は、プリフェッチバッファに保持されているデータをプロセッサに転送するデータ転送手段とを備えたものである。

【0012】請求項1記載のプリフェッチ装置によれば、プロセッサがプリフェッチ空間に対しリード要求を出すことにより、プリフェッチ装置はプロセッサの動作

とは無関係に主記憶からデータを読み出し、プリフェッチバッファに保持する。その後、プロセッサがプリフェッチ空間に対しリード要求を出したとき、そのアドレスがプリフェッチバッファに保持されているデータのアドレスであれば、主記憶からの読み出しを行わず、代わりにプリフェッチバッファに格納されているデータをプロセッサに転送する。すなわち、プロセッサのプリフェッチ空間へのリード要求のアドレスが、プリフェッチ装置で想定しているプリフェッチアドレスと一致する場合は、プロセッサのプリフェッチ空間へのリード要求の待ち時間が生じないため、高速なプログラムの実行を行うことが出来る。

【0013】また、プリフェッチ空間へのリード要求を出す命令は、キャッシュ空間や通常空間へのリード要求を出す命令と同じ命令であるため、コンパイラでプリフェッチ空間と他の空間を区別して命令生成する必要がないという効果がある。

【0014】さらに、プリフェッチ空間へのリード要求を出す命令を実行することではキャッシュメモリは動作せず、キャッシュメモリに保持されているデータは変化することはない。したがって、ストリームデータなどのプリフェッチによるキャッシュメモリのヒット率の低下を招くことはない。

【0015】請求項2記載のプリフェッチ装置は、請求項1において、プリフェッチ手段におけるデータの読み出しを繰り返す反復プリフェッチ手段と、この反復プリフェッチ手段を終了させるプリフェッチ終了手段とを備えたものである。

【0016】請求項2記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0017】請求項3記載のプリフェッチ装置は、請求項2において、反復プリフェッチ手段の行われた回数を計数する回数計数手段と、制御レジスタに設定された値と回数計数手段により計数されたプリフェッチ手段の回数とが一致することを検出する回数一致手段と、この回数一致手段により一致を検出した場合は反復プリフェッチ手段を終了する手段を備えたものである。

【0018】請求項3記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0019】請求項4記載のプリフェッチ装置は、請求項2において、反復プリフェッチ手段によりプリフェッチバッファに保持したデータの大きさを計数するデータサイズ計数手段と、制御レジスタに設定されたデータサイズとデータサイズ計数手段により得られたプリフェッチバッファに保持されているデータの大きさとが一致することを検出するデータサイズ一致検出手段と、このデータサイズ一致検出手段により一致が検出された場合は反復プリフェッチ手段を終了する手段を備えたものである。

【0020】請求項4記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

ば、請求項1と同様な効果がある。

【0021】請求項5記載のプリフェッチ装置は、請求項2において、反復プリフェッチ手段のアドレス生成手段により生成したアドレスが制御レジスタに設定されたアドレスと一致することを検出する終了アドレス検出手段と、この終了アドレス検出手段により一致が検出された場合は、反復プリフェッチ手段を終了する手段を備えたものである。

【0022】請求項5記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0023】請求項6記載のプリフェッチ装置は、請求項1において、アドレス生成手段が、プロセッサの記憶デバイスへのリード要求アドレスに制御レジスタに設定された値を加算することでアドレスを生成する手段である。

【0024】請求項6記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0025】請求項7記載のプリフェッチ装置は、請求項1において、アドレス生成手段が、プロセッサの記憶デバイスへのリード要求アドレスに対し、制御レジスタに設定された情報により決定される演算を施すことでアドレスを生成する手段である。

【0026】請求項7記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0027】請求項8記載のプリフェッチ装置は、請求項3から請求項7のいずれかの1項において、プロセッサにより制御レジスタを書き換える手段を備えたものである。

【0028】請求項8記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0029】請求項9記載のプリフェッチ装置は、請求項1において、プリフェッチ手段におけるデータの読み出しを繰り返す第1と第2の反復プリフェッチ手段と、プロセッサの記憶デバイスへの第1のリード要求アドレスにより第1の反復プリフェッチ手段が開始され、プロセッサの記憶デバイスへの第2のリード要求アドレスと第1の反復プリフェッチ手段によりプリフェッチバッファに保持されているデータのアドレスとが一致していないことがアドレス一致検出手段により検出された場合は、第1の反復プリフェッチ手段を終了し、第2の反復プリフェッチ手段を開始する手段を備えたものである。

【0030】請求項9記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0031】請求項10記載のプリフェッチ装置は、請求項1において、データ転送手段によりプリフェッチバッファのデータをプロセッサに転送した場合は、プリフェッチバッファに保持されているデータをプリフェッチバッファから無効化する無効化手段を備えたものである。

【0032】請求項10記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

ば、請求項1と同様な効果がある。

【0033】請求項1記載のプリフェッチ装置は、請求項1において、アドレス一致検出手段によりプロセッサの記憶デバイスへのリード要求アドレスとプリフェッチバッファに保持したデータのアドレスとが一致しないことを検出した場合は、プリフェッチバッファに保持されたデータを無効化する手段を備えたものである。

【0034】請求項1記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0035】請求項1記載のプリフェッチ方法は、プロセッサの記憶デバイスへのリード要求アドレスがプリフェッチ空間であることを判別する空間判別ステップと、プリフェッチアドレスを生成するアドレス生成ステップと、このアドレス生成ステップで得たプリフェッチアドレスを用いて記憶デバイスから対応するデータを読み出すデータ読み出しステップと、プロセッサの記憶デバイスへのリード要求アドレスが空間判別ステップによりプリフェッチ空間であることが判別された場合はプロセッサの記憶デバイスへのリード要求とは無関係にデータ読み出しステップによりデータを読み出し、プリフェッチバッファに保持するプリフェッチステップと、プロセッサの記憶デバイスへのリード要求アドレスがプリフェッチバッファに保持されているデータのアドレスと一致することを検出するアドレス一致検出ステップと、このアドレス一致検出ステップにより一致が検出された場合は、プリフェッチバッファに保持されているデータをプロセッサに転送するデータ転送ステップとを含むものである。

【0036】請求項1記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0037】請求項1記載のプリフェッチ方法は、請求項1において、プリフェッチステップにおけるデータの読み出しを繰り返す反復プリフェッチステップと、この反復プリフェッチステップを終了させるプリフェッチ終了ステップとを含むものである。

【0038】請求項1記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0039】請求項1記載のプリフェッチ方法は、請求項1において、反復プリフェッチステップの行われた回数を計数する回数計数ステップと、制御レジスタに設定された値と回数計数ステップにより計数されたプリフェッチステップの回数とが一致することを検出する回数一致ステップと、回数一致ステップにより一致を検出した場合は反復プリフェッチステップを終了するステップを含むものである。

【0040】請求項1記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0041】請求項1記載のプリフェッチ方法は、請求項1において、反復プリフェッチステップによりプリフェッチバッファに保持したデータの大きさを計数す

るデータサイズ計数ステップと、制御レジスタに設定されたデータサイズとデータサイズ計数ステップにより得られたプリフェッチバッファに保持されているデータの大きさとが一致することを検出するデータサイズ一致検出ステップと、このデータサイズ一致検出ステップにより一致が検出された場合は反復プリフェッチステップを終了するステップを含むものである。

【0042】請求項1記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0043】請求項1記載のプリフェッチ方法は、請求項1において、反復プリフェッチステップのアドレス生成ステップにより生成したアドレスが制御レジスタに設定されたアドレスと一致することを検出する終了アドレス検出ステップと、この終了アドレス検出ステップにより一致が検出された場合は、反復プリフェッチステップを終了するステップを含むものである。

【0044】請求項1記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0045】請求項1記載のプリフェッチ方法は、請求項1において、アドレス生成ステップが、プロセッサの記憶デバイスへのリード要求アドレスに制御レジスタに設定された値を加算することでアドレスを生成するステップである。

【0046】請求項1記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0047】請求項1記載のプリフェッチ方法は、請求項1において、アドレス生成ステップが、プロセッサの記憶デバイスへのリード要求アドレスに対し、制御レジスタに設定された情報により決定される演算を施すことでアドレスを生成するステップである。

【0048】請求項1記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0049】請求項1記載のプリフェッチ方法は、請求項1から請求項18のいずれかの1項において、プロセッサにより制御レジスタを書き換えるステップを含むものである。

【0050】請求項1記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0051】請求項20記載のプリフェッチ方法は、請求項1において、プリフェッチステップにおけるデータの読み出しを繰り返す第1と第2の反復プリフェッチステップと、プロセッサの記憶デバイスへの第1のリード要求アドレスにより第1の反復プリフェッチステップが開始され、プロセッサの記憶デバイスへの第2のリード要求アドレスと第1の反復プリフェッチステップによりプリフェッチバッファに保持されているデータのアドレスとが一致していないことがアドレス一致検出ステップにより検出された場合は、第1の反復プリフェッチステップを終了し、第2の反復プリフェッチステップを開始するステップを含むものである。



【0052】請求項20記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0053】請求項21記載のプリフェッチ方法は、請求項12において、データ転送ステップによりプリフェッチバッファのデータをプロセッサに転送した場合が、プリフェッチバッファに保持されているデータをプリフェッチバッファから無効化する無効化ステップを含むものである。

【0054】請求項21記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0055】請求項22記載のプリフェッチ方法は、請求項12において、アドレス一致検出ステップによりプロセッサの記憶デバイスへのリード要求アドレスとプリフェッチバッファに保持したデータのアドレスとが一致しないことを検出した場合は、プリフェッチバッファに保持されたデータを無効化するステップを含むものである。

【0056】請求項22記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0057】

【発明の実施の形態】以下、本発明の第1の実施の形態のプリフェッチ方法についてプリフェッチ装置を用いて、図1から図6により説明する。図1は本発明の第1の実施の形態におけるプリフェッチ装置を含むシステムのハードウェア構成図を示すブロック図である。1はプリフェッチ装置、2は記憶デバイスとしての主記憶、3は命令が格納された命令メモリ、4は命令メモリ3に格納された命令にしたがって処理を行い必要に応じて主記憶2に対しデータを読み出す要求を行う機能をはじめ通常のプロセッサとしての機能を有するプロセッサ、5はプロセッサ4が主記憶2から読み出したデータのコピーを格納しておく機能をはじめ通常のキャッシュメモリとしての機能を有するキャッシュメモリである。

【0058】プリフェッチ装置1において、10はプロセッサ4の出力したアドレスを参照してアドレス空間を判別するアドレス空間判別回路であり、請求項記載の空間判別手段に対応する。

【0059】20はプロセッサ4の出力したアドレスを元にプリフェッチアドレスを生成するプリフェッチアドレス生成回路であり、請求項記載のアドレス生成手段に対応する。図6に本発明の第1の実施の形態におけるプリフェッチアドレス生成回路20のブロック図を示している。21は次のプリフェッチアドレスの生成に用いるアドレスを保持するバッファであり、22はプロセッサ4の出力したアドレスとプリフェッチアドレス生成回路20の出力したアドレスとを選択しバッファ21に出力するマルチプレクサであり、23はバッファ21に保持されたアドレスとプロセッサ4のアドレスを加算するための加算器である。

【0060】図1に戻って、30はプリフェッチにより

主記憶2から読み出されたデータを保持しておくプリフェッチバッファであり、請求項記載のデータ読み出し手段とプリフェッチバッファ30とでプリフェッチ手段に対応する。

【0061】40はプリフェッチアドレス生成回路20の出力したアドレスを保持しておくプリフェッチアドレスバッファであり、50はプロセッサ4の出力したアドレスとプリフェッチアドレスバッファ40に格納されたアドレスとを比較し一致信号をプリフェッチ制御回路100に出力する比較回路であり、プリフェッチアドレスバッファ40と比較回路50で請求項記載のアドレス一致検出手段に対応する。比較回路50により一致が検出された場合は、データ転送手段によりプリフェッチバッファ30に保持されているデータをプロセッサ4に転送し、データ転送手段によりプリフェッチバッファ30のデータをプロセッサ4に転送した場合は、無効化手段によりプリフェッチバッファ30に保持されているデータをプリフェッチバッファから無効化する。また、データ転送手段は図1のマルチプレクサ102が対応し、無効化手段はプリフェッチアドレスバッファ40とプリフェッチ制御回路100が対応する。すなわち、プリフェッチ制御回路100の内部状態がプリフェッチ状態のときは無効であることをあらわす。

【0062】100はプリフェッチ装置1の全体を制御するプリフェッチ制御回路であり、内部状態として図2の状態遷移図に示す「非プリフェッチ状態」、「プリフェッチ状態」、「プリフェッチ待ち状態」の3状態を有する。「非プリフェッチ状態」はプリフェッチを行わない状態であり、「プリフェッチ状態」はプロセッサ4の読み出し要求アドレスがプリフェッチ空間であると判断された場合に「非プリフェッチ状態」から遷移し、「プリフェッチ状態」である間はプリフェッチを行う。「プリフェッチ待ち状態」はプリフェッチバッファ30にデータが保持されておりこれ以上プリフェッチを行うことができない状態である。プリフェッチバッファ30にはデータが1つしか保持することができないため「プリフェッチ状態」でデータを読み出すと「プリフェッチ待ち状態」に遷移する。また、プリフェッチバッファ30のデータがプロセッサ4に転送されるとプリフェッチバッファ30のデータは無効化されるので状態は「プリフェッチ状態」に戻る。

【0063】図1に戻って、101はプロセッサ4の出力するアドレスとプリフェッチアドレス生成回路20の出力するアドレスとを選択し、主記憶2へ出力するマルチプレクサであり、102は主記憶2の出力するデータとプリフェッチバッファ30の出力するデータとを選択し、プロセッサ4へ出力するマルチプレクサである。

【0064】図3は本発明のプリフェッチ装置1を備えたシステムにおけるアドレス空間マップを示している。本実施の形態のシステムにおけるアドレスは32ビット



の幅であるとしている。全アドレス空間は3つの空間、「通常空間」、「プリフェッチ空間」、「キャッシュ空間」に分割されており、通常空間はアドレス0x00000000~0x3FFFFFFFに割り当てられ、プリフェッチ空間はアドレス0x40000000~0x7FFFFFFFに割り当てられ、キャッシュ空間はアドレス0x80000000~0xFFFFFFFFに割り当てられている。すなわち、アドレスの上位2ビットを参照するだけで属している空間を判別することができる。なお、“0x”は16進数であることを示す接頭辞である。

【0065】「プリフェッチ空間」にはプロセッサ4により連続的にアクセスされ、プロセッサ4が一度しか使用しないデータを配置する。これは例えば、ストリームデータのように、プロセッサ4の外部で生成され主記憶2に格納されるデータで、プロセッサ4が読み出すと主記憶2上のデータは新しいデータに置き換わってしまうデータを想定している。第1の実施の形態ではプロセッサ4が「プリフェッチ空間」へリード要求を出すことによりプリフェッチ装置1が動作する。

【0066】「キャッシュ空間」にはプロセッサ4が再利用可能なデータを配置する。「キャッシュ空間」に配置されたデータはプロセッサ4が一度読み出すと、キャッシュメモリ5にコピーされ、次回からキャッシュメモリ5のデータがプロセッサ4に供給され、主記憶2のデータは読み出されない。また、プリフェッチ装置1は動作しない。

【0067】なお、プロセッサ4が「通常空間」へのアクセスを行った場合は、プリフェッチ装置1およびキャッシュメモリ5は動作せず直接主記憶2がアクセスされる。第1の実施の形態での動作説明には「通常空間」は使用しないが、実際のプリフェッチ装置1を含むシステムにおいては、「通常空間」のようなアドレス空間が必要である。

【0068】図4は本実施の形態で説明に用いるプログラム例を示している。プログラム例中の命令i0~i10は命令メモリ3に格納されており、プロセッサ4によって順次読み出され実行される。

【0069】以下、各命令について説明する。命令i0、i1、i2はそれぞれ0x40000100、0x80000100、256をレジスタr0、r1、r2に格納する命令であり、それぞれ命令i3のロード用アドレス、命令i4のロード用アドレス、ループカウンタとして使用される。

【0070】命令i3はレジスタr0の値をアドレスとして主記憶2から4バイトのデータを読み出し、レジスタr3に格納する命令であり、命令i4はレジスタr1の値をアドレスとして主記憶2から4バイトのデータを読み出し、レジスタr4に格納する命令である。

【0071】命令i5はレジスタr3の値と、レジスタ

r4の値とを乗算し、その結果を積和レジスタMの値に加算し、レジスタMに格納する積和命令である。レジスタMは暗黙に指定されているものとする。

【0072】命令i6、i7はそれぞれレジスタr0、r1に4を加算する命令である。レジスタr0、r1はそれぞれ命令i3、i4で4バイトデータのアドレスとして使われており、次の4バイトデータのアドレスに更新するため4を加算しているものである。本プログラム例はループ構造をなしているので、命令i3、命令i4は繰り返し実行されることになる。したがって、0x40000100および0x80000100の後続のデータは4バイト単位で連続してアクセスされることになる。

【0073】命令i8はレジスタr2に-1を加算する命令で、命令i9はレジスタr2と0とを比較する命令であり、命令i10は命令i9の比較結果でレジスタr2の方が大きい場合、ラベル“loop”に分岐し、そうでない場合分岐しない命令である。レジスタr2は本プログラム例で構成されているループの回数を数数しているカウンタとして使用されており、命令i3~i10で256回のループを構成している。この場合、プリフェッチ手段はデータの読み出しを繰り返す反復プリフェッチ手段と、この反復プリフェッチ手段を終了させるプリフェッチ終了手段により構成されることとなる。

【0074】図5は図4に示したプログラム例をプロセッサ4が実行した場合の動作波形を示した図である。C0~C20は各サイクルを表しており、C0~C10は図4のプログラム例の命令i3~i10で構成されるループの1回目を示しており、C11~C18は2回目のループを示している。

【0075】図5において、「プロセッサの実行命令」の波形は各サイクルのプロセッサ4で実行されている命令を表しており、「プロセッサのアドレス」はプロセッサ4が出力するアドレスを表している。「プロセッサからプリフェッチ装置」の「リード要求」はプロセッサ4のプリフェッチ装置1に対するリード要求信号であり、「アクノレッジ」はプリフェッチ装置1のプロセッサ4に対するアクノレッジ信号を表している。それぞれの信号はハイ状態のときアクティブ状態であるとしている。

【0076】同様に「プリフェッチ装置から主記憶」の「リード要求」はプリフェッチ装置1の主記憶2に対するリード要求信号であり、「アクノレッジ」は主記憶2のプリフェッチ装置1に対するアクノレッジ信号を表している。同様にそれぞれの信号はハイ状態のときアクティブ状態であるとしている。

【0077】「主記憶」の波形は主記憶2の読み出し動作を表しており、波形図中の値は読み出すときのアドレスを表している。なお、第1の実施の形態では主記憶2の「読み出し」は4サイクルで完了するとしている。

【0078】「プリフェッチアドレスバッファ」の波形

はプリフェッチアドレスバッファ40に保持されているアドレスを表しており、「プリフェッチバッファ」の波形はプリフェッチバッファ30に保持されているデータを表している。プリフェッチバッファの波形図中の値はアドレスであり、そのアドレスで読み出したデータが保持されていることを示している。なお、黒く塗りつぶした部分は保持されているデータが無効であることを示している。

【0079】「プリフェッチ制御回路の内部状態」の波形は図2で示したプリフェッチ制御回路100の内部状態を示している。

【0080】以下、図4のプログラム例における第1の実施の形態のプリフェッチ装置を含むシステムの動作について説明する。

【0081】まず、プロセッサ4は命令i0～i2を実行し、レジスタr0、r1、r2にそれぞれ0x40000100、0x80000100、256を格納する。この部分は図5には示されていない。

【0082】次に、プロセッサ4は命令i3を実行する。命令i3はレジスタr0に格納されている値0x40000100をアドレスとして主記憶2からデータを読み出す命令である。図5のC0にプロセッサ4は命令i3を実行している様子を示している。命令i3により、プロセッサ4はリード要求信号をプリフェッチ装置1に出力すると同時に、アドレス0x40000100をマルチプレクサ101に出力する。プリフェッチ装置1においては、プロセッサ4の出力したリード要求信号はプリフェッチ制御回路100に入力される。このとき、リード要求アドレスのアドレス空間はキャッシュ空間と異なるのでキャッシュメモリ5は動作しない。

【0083】プリフェッチ制御回路100ではプロセッサ4から出力されたリード要求信号によりマルチプレクサ101を制御することでプロセッサ4の出力したアドレスを選択し主記憶2へ送ると同時に、主記憶2に対しリード要求信号を出力する。

【0084】一方、アドレス空間判別回路10はプロセッサ4の出力するアドレスが0x40000100であることによりプリフェッチ空間であると判断し、その旨の信号をプリフェッチ制御回路100に出力する。プリフェッチ制御回路100はアドレス空間判別回路10の出力する信号と、プロセッサ4のリード要求信号により内部状態を「プリフェッチ状態」に移行させるとともに、プリフェッチアドレス生成回路20を制御し、プロセッサ4の出力するアドレスをバッファ21（図6）に保持させる。

【0085】C1～C2は主記憶2の読み出しを待っているサイクルである。主記憶2の読み出しは4サイクルであり、命令i3はデータの読み出しが完了するまで次の命令に進むことができないので、命令i3が実行中の状態となっている。

【0086】C3では、主記憶2はアドレス0x40000100のデータを読み出しマルチプレクサ102に出力するとともに、アクノレッジ信号をプリフェッチ制御回路100に出力する。プリフェッチ制御回路100はマルチプレクサ102を制御し、主記憶2の出力したデータをプロセッサ4に出力し、アクノレッジ信号をプロセッサ4に出力する。キャッシュメモリ5は動作していないので主記憶2の出力しているデータがキャッシュメモリ5に取り込まれることはない。

【0087】C4ではプロセッサ4は命令i4を実行する。命令i4はld命令であるが、アドレスが0x80000100、すなわちキャッシュ空間へのリード要求であるためキャッシュメモリ5が動作する。ここではキャッシュメモリ5はヒットするとしているので、プロセッサ4はデータ待ちになることはなく、命令i4を完了させることができ、直ちに命令i5の実行を開始できる。キャッシュメモリ5がヒットするとしているので、プリフェッチ装置1および主記憶2へのアクセスは行われず、動作しない。

【0088】一方、プリフェッチ制御回路100の内部状態は「プリフェッチ状態」になっているので、主記憶2に対しプリフェッチ動作を開始する。プリフェッチ動作はプロセッサ4の動作とは無関係に行われる。まず、プリフェッチ制御回路100はプリフェッチアドレス生成回路20に対し、最初のプリフェッチアドレスを生成するように制御する。プリフェッチアドレス生成回路20は、バッファ21に保持したアドレス0x40000100と値4を加算器23で加算した値0x40000104をマルチプレクサ101に出力する。さらに、マルチプレクサ101を制御し、プリフェッチアドレス生成回路20の出力するアドレス0x40000104を主記憶2に出力するようにし、同時にリード要求信号を主記憶2に出力する。主記憶2ではアドレス0x40000104のデータを4サイクルかけて読み出す。

【0089】C5、C6ではプロセッサ4は命令i5、命令i6を実行しているが、この間主記憶2へのアクセスは行わない。

【0090】C7では、主記憶2はプリフェッチ動作を完了し、読み出したデータをプリフェッチバッファ30およびマルチプレクサ102に出力し、アクノレッジ信号をプリフェッチ制御回路100に出力する。プリフェッチ制御回路100は主記憶2のアクノレッジ信号により、プリフェッチバッファ30に主記憶2のデータを保持させ、またプリフェッチアドレスバッファ40にプリフェッチアドレス生成回路20が出力するアドレスを保持させる。さらに、プリフェッチ制御回路100は内部状態を「プリフェッチ状態」から「プリフェッチ待ち状態」に変化させる。また、プリフェッチアドレス生成回路20では現在出力しているアドレス0x40000104を内部に保持し直し、4を加算した0x40000

108を出力する。これらの動作はプロセッサ4の命令実行に影響を与えることはない。

【0091】C8～C10ではプロセッサ4は命令i8～i10を実行しているが、この間主記憶2へのアクセスは行わない。

【0092】命令i6～i8によりプロセッサ4のレジスタr0、r1、r2の値はそれぞれ0x40000104、0x80000100、255となる。命令i9によりレジスタr2と値0が比較されるが、レジスタr2の方が「大」なので命令i10により、ラベル「loop」への分岐が成立し、1回目のループが終了する。次のC11からは命令i3が実行されることになる。

【0093】C11ではプロセッサ4は再び命令i3を実行する。命令i3はld命令であるのでプロセッサ4はレジスタr0の値である0x40000104をアドレスとして出力し、リード要求信号をプリフェッチ制御回路100に出力する。ここで、プリフェッチアドレスバッファ40にはアドレス0x40000104が保持されており、比較回路50はプロセッサ4の出力するアドレスと比較し一致することを検出し、一致信号をプリフェッチ制御回路100に出力する。プリフェッチ制御回路100はプロセッサ4のリード要求信号と、比較回路50の一致信号により、マルチプレクサ102を制御し、プリフェッチバッファ30に保持されているアドレス0x40000104のデータをプロセッサ4に出力するとともに、アクノレッジ信号をプロセッサ4に出力する。この場合の命令i3のld命令は1サイクル(0サイクル待ち)で完了したことになる。このとき、リード要求アドレスのアドレス空間はキャッシュ空間と異なるのでキャッシュメモリ5は動作しない。

【0094】一方、プリフェッチバッファ30に保持されていたデータはプロセッサ4への転送が完了したので無効化され、プリフェッチ制御回路100は内部状態を「プリフェッチ待ち状態」から「プリフェッチ状態」に移す。

【0095】C12ではプロセッサ4は命令i4を実行する。命令i4はld命令であるが、アドレスが0x80000104、すなわちキャッシュ空間へのリード要求であるためキャッシュメモリ5が動作する。ここではキャッシュメモリ5はヒットするとしているので、プロセッサ4はデータ待ちになることはなく、命令i4を完了させることができ、直ちに命令i5の実行を開始できる。キャッシュメモリ5がヒットするとしているので、プリフェッチ装置1および主記憶2へのアクセスは行われず、動作しない。

【0096】一方、プリフェッチ制御回路100の内部状態は「プリフェッチ状態」になっているので、プロセッサ4の命令動作とは無関係に主記憶2に対するプリフェッチ動作を開始する。プリフェッチ制御回路100は主記憶2に対しリード要求信号を出力すると同時に、マ

ルチプレクサ101を制御してプリフェッチアドレス生成回路20の出力しているアドレス0x40000108を主記憶2に出力する。

【0097】C13以降は、1回目のループであるC5以降と同様に動作する。特に、プリフェッチバッファ30に保持されるデータは、その後プロセッサ4が命令i3を実行するときに、主記憶2の読み出しを行うことなしにプロセッサ4に転送され、プロセッサ4は命令i3の完了を待たされることがない。

【0098】以上のように本発明の実施の形態のプリフェッチ装置によれば、プロセッサ4の1回目の命令i3の実行で、プリフェッチ空間に対しリード要求を出すことにより、プリフェッチ装置1はプロセッサ4の動作とは無関係に主記憶2から後続アドレスデータを読み出し、プリフェッチバッファ30に保持する。その後、プロセッサ4のリード要求で、そのアドレスのデータがプリフェッチバッファ30に保持されている場合は、主記憶2のアクセスを行うことなしにプリフェッチバッファ30からプロセッサ4に送る。そのため、プロセッサ4は主記憶2の読み出しを待つことがなく、高速な読み出しを行うことができる。

【0099】さらに、図4のプログラムのようにプリフェッチ空間へのリード要求のアドレスが、プリフェッチ装置1で想定しているプリフェッチアドレスと一致する場合は、プロセッサ4のプリフェッチ空間へのリード要求の待ち時間が生じないため、ストリームフィルタのようなプログラムを高速に実行することができる。

【0100】また、本発明の第1の実施の形態によれば、命令i3、命令i4は同じld命令であり、コンパイラでプリフェッチ空間へのアクセスとキャッシュ空間へのアクセスを区別することなく命令を生成することができる。

【0101】さらに、命令i3を実行することではキャッシュメモリ5は動作せず、キャッシュメモリ5に保持されているデータは変化しない。そのため、プリフェッチによるキャッシュメモリのヒット率低下を招くことはない。

【0102】ここで、例えば命令i3がキャッシュ空間へのアクセス命令であり、命令i3のアドレスと命令i4のアドレスがキャッシュメモリ5上で同一のエントリを占める場合は、命令i3の実行により生じたキャッシュミスにより、命令i4で読み出すデータはキャッシュメモリ5から追い出されてしまい命令i4実行時は必ずミスヒットとなる。また、次の機会で本実施の形態のプログラム例が実行されるときは、命令i3のデータは外部で生成され主記憶2上に格納されるため、キャッシュメモリ5に残っているデータを破棄しなければならない。このように、命令i3のデータをキャッシュメモリ5に保持させることは無駄であるばかりでなく、命令i4のようなキャッシュメモリ5に保持させると有効な

データまで追い出してしまう恐れがある。本発明の実施の形態によれば命令13によりキャッシュメモリ5は動作せず、キャッシュメモリ5と同等の動作をすることが可能である。

【0103】なお、上記の動作説明において、例えばC11でプロセッサ4のリード要求アドレスとプリフェッチアドレスバッファ40に保持されているアドレスとが一致せず、かつそのアドレスがプリフェッチ空間内にあることが比較回路50で検出された場合は、プリフェッチバッファ30のデータを破棄し、再度主記憶2からプロセッサ4の要求するデータを読み出し、プリフェッチバッファ30に保持すると同時に、そのときのアドレスをプリフェッチアドレスバッファ40に保持する。プロセッサ4の出力したアドレスはプリフェッチアドレス生成回路20のバッファ21にも保持され、その後のプリフェッチに用いられる。このようにプリフェッチ装置1の想定するアドレスと異なるプリフェッチ空間のアドレスをプロセッサ4が出力した場合でも誤動作することはない。

【0104】これは、プリフェッチバッファ30におけるデータの読み出しを繰り返す第1と第2の反復プリフェッチ手段と、プロセッサ4の主記憶2への第1のリード要求アドレスにより第1の反復プリフェッチ手段が開始され、プロセッサ4の主記憶2への第2のリード要求アドレスと第1の反復プリフェッチ手段によりプリフェッチバッファに保持されているデータのアドレスとが一致していないことが比較回路50により検出された場合は、第1の反復プリフェッチ手段を終了し、第2の反復プリフェッチ手段を開始する手段により構成されている。

【0105】なお、本発明の実施の形態におけるプリフェッチアドレス生成回路20は図6に示すような4を加算する例を示しているが、4以外の他の値でも良く、また加算以外の他の演算でも同様の効果を得ることができる。

【0106】またアドレス一致検出手段(50)によりプロセッサ4の記憶デバイス(2)へのリード要求アドレスとプリフェッチバッファ30に保持したデータのアドレスとが一致しないことを検出した場合は、プリフェッチバッファ30に保持されたデータを無効化する手段を設けるようにしてもよい。

【0107】また、プリフェッチアドレス生成回路の他の実施の形態として、図7に示すようにプロセッサ4で書換え可能な制御レジスタ25により演算器24の演算方法とオペランドを指定できるようにしてもよい。図7において、24はオペランドに対して複数の演算を施すことのできる演算器であり、25は演算器24に与える演算コマンドおよびオペランドを保持し、プロセッサ4により書き換え可能な制御レジスタである。図7のプリフェッチアドレス生成回路により、例えば、下位4ビット

トのビット列を逆転する演算の場合と4を加算する場合とを制御レジスタ25により切り替えるようにすると、プロセッサ4で実行するプログラムの特性にあわせてプリフェッチのアドレス生成が容易になり本発明の効果はより高くなる。

【0108】また、図8に示すようにテーブルから取り出したオフセットをアドレスに加算してプリフェッチアドレスを生成するようにしても良い。図8において、26はカウンタであり、27はカウンタ26の値をインデックスとしてオフセットを読み出すことのできるテーブルROMであり、28はテーブルROM27から読み出したオフセットとレジスタ21に格納されたアドレスとを加算することでアドレスを生成する加算器である。図8のプリフェッチアドレス生成回路20により、例えば、テーブルROM27にランダムなオフセットを格納しておくだけで、ランダムなアドレスを生成することもできる。

【0109】さらに、図9にプリフェッチを終了させる回路を備えた他の実施の形態のプリフェッチ装置のブロック図を示す。200は終了アドレスを保持する制御レジスタであり、201はプリフェッチアドレス生成回路20の出力するアドレスと制御レジスタ200に保持されているアドレスとを比較する比較回路であり、請求項5の終了アドレス検出手段に対応する。この終了アドレス検出手段により一致が検出された場合は、終了手段により反復プリフェッチ手段を終了する。6は制御レジスタ200と比較回路201を備えたプリフェッチ装置であり、103はプリフェッチ装置6の全体を制御するプリフェッチ制御回路である。図9のプリフェッチ装置6、プリフェッチ制御回路103はそれぞれ図1のプリフェッチ装置1、プリフェッチ制御回路100と同様に動作するが、比較回路201によりプリフェッチアドレス生成回路20の出力するアドレスと制御レジスタ200に保持されているアドレスとが一致することが検出された場合は、その旨の信号をプリフェッチ制御回路103に出力する。プリフェッチ制御回路103は内部状態を「非プリフェッチ状態」に遷移させプリフェッチ動作を終了する。図9の図1と同一符号で表した部分は図1と共通しその説明を省略している。

【0110】図9のプリフェッチ装置6を含むシステムは、プロセッサ4があらかじめ制御レジスタ200に終了アドレスを設定しておき、プリフェッチ動作開始後、プリフェッチ動作で用いるアドレスが制御レジスタ200に設定されたアドレスと等しい場合はプリフェッチ動作を終了する。これによりプロセッサ4が使うことのない無駄なプリフェッチ動作を無くすることができるという効果がある。

【0111】なお、図9では終了アドレスを指定してプリフェッチアドレスと一致する場合に終了する回路を示したが、プリフェッチの回数を計数し、制御レジスタで

指定した回数に達した場合に終了する手段でも同様の効果を得ることができる。この場合、反復プリフェッチ手段の行われた回数を計数する回数計数手段と、制御レジスタに設定された値と回数計数手段により計数されたプリフェッチ手段の回数とが一致することを検出する回数一致手段と、この回数一致手段により一致を検出した場合は反復プリフェッチ手段を終了する手段により構成される。また、プリフェッチしたトータルのデータサイズを計数し、制御レジスタで指定したデータサイズに達した場合に終了する手段でも同様の効果を得ることができる。この場合、反復プリフェッチ手段によりプリフェッチバッファに保持したデータの大きさを計数するデータサイズ計数手段と、制御レジスタに設定されたデータサイズとデータサイズ計数手段により得られたプリフェッチバッファに保持されているデータの大きさとが一致することを検出するデータサイズ一致検出手段と、このデータサイズ一致検出手段により一致が検出された場合は反復プリフェッチ手段を終了する手段により構成される。

【0112】なお、図1および図9のプリフェッチバッファ30は主記憶2から読み出したデータを1つだけ保持する構成を例示しているが、複数のデータを保持できる構成の方がよりプリフェッチ動作の効果が高くなる。例えば、4つ分のデータを保持できる構成の場合は、何らかの要因で主記憶2がアクセスできない場合でもプロセッサ4へのデータ供給にある程度応えることができる。

【0113】なお、主記憶2は記憶デバイスであればよく、例えば他のデバイスとして、ハードディスクのような記録メディアなどでも同様の効果を得ることができることは言うまでもない。

【0114】

【発明の効果】請求項1記載のプリフェッチ装置によれば、プロセッサがプリフェッチ空間に対しリード要求を出すことにより、プリフェッチ装置はプロセッサの動作とは無関係に主記憶からデータを読み出し、プリフェッチバッファに保持する。その後、プロセッサがプリフェッチ空間に対しリード要求を出したとき、そのアドレスがプリフェッチバッファに保持されているデータのアドレスであれば、主記憶からの読み出しを行わず、代わりにプリフェッチバッファに格納されているデータをプロセッサに転送する。すなわち、プロセッサのプリフェッチ空間へのリード要求のアドレスが、プリフェッチ装置で想定しているプリフェッチアドレスと一致する場合は、プロセッサのプリフェッチ空間へのリード要求の待ち時間が生じないため、高速なプログラムの実行を行うことができる。

【0115】また、プリフェッチ空間へのリード要求を出す命令は、キャッシュ空間や通常空間へのリード要求を出す命令と同じ命令であるため、コンパイラでプリフ

ェッチ空間と他の空間を区別して命令生成する必要がないという効果がある。

【0116】さらに、プリフェッチ空間へのリード要求を出す命令を実行することではキャッシュメモリは動作せず、キャッシュメモリに保持されているデータは変化することはない。したがって、ストリームデータなどのプリフェッチによるキャッシュメモリのヒット率の低下を招くことはない。

【0117】請求項2記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0118】請求項3記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0119】請求項4記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0120】請求項5記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0121】請求項6記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0122】請求項7記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0123】請求項8記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0124】請求項9記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0125】請求項10記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0126】請求項11記載のプリフェッチ装置によれば、請求項1と同様な効果がある。

【0127】請求項12記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0128】請求項13記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0129】請求項14記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0130】請求項15記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0131】請求項16記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0132】請求項17記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0133】請求項18記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0134】請求項19記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0135】請求項20記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0136】請求項21記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

【0137】請求項22記載のプリフェッチ方法によれば、請求項1と同様な効果がある。

## 【図面の簡単な説明】

【図1】本発明の第1の実施の形態におけるプリフェッチ装置を含むシステムのハードウェア構成図である。

【図2】第1の実施の形態におけるプリフェッチ制御回路100の内部状態遷移を示す説明図である。

【図3】第1の実施の形態におけるプリフェッチ装置を含むシステムの動作説明用のアドレス空間マップ図である。

【図4】第1の実施の形態におけるプリフェッチ装置を含むシステムの動作説明用のプログラム例の説明図である。

【図5】第1の実施の形態におけるプリフェッチ装置を含むシステムの動作説明用の動作波形図である。

【図6】第1の実施の形態におけるプリフェッチアドレス生成回路20の詳細ブロック図である。

【図7】本発明の他の実施の形態におけるプリフェッチアドレス生成回路20のブロック図である。

【図8】本発明のさらに他の実施の形態におけるプリフェッチアドレス生成回路20のブロック図である。

【図9】本発明のさらに他の実施の形態におけるプリフェッチ装置を含むシステムのハードウェア構成図である。

【図10】従来のプリフェッチ命令によるプリフェッチ装置の動作説明用のプログラム動作説明図である。

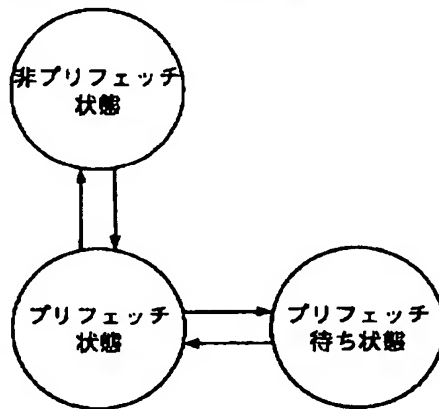
【図11】従来のキャッシュメモリによるプリフェッチ装置の動作説明用のプログラム動作説明図である。

## 【符号の説明】

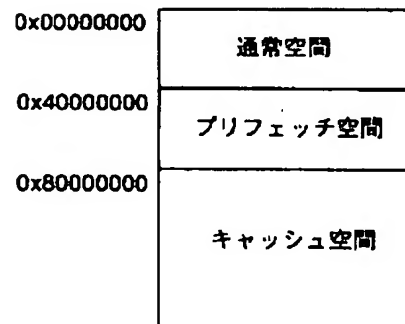
1	プリフェッチ装置
2	主記憶
3	命令メモリ
4	プロセッサ
5	キャッシュメモリ
10	アドレス空間判別回路
20	プリフェッチアドレス生成回路
21	バッファ
22	マルチプレクサ
23	加算器
24	演算器
25	制御レジスタ
30	プリフェッチバッファ
40	プリフェッチアドレスバッファ
50	比較回路
100	プリフェッチ制御回路
101	マルチプレクサ
102	マルチプレクサ
200	制御レジスタ
201	比較回路

【図2】

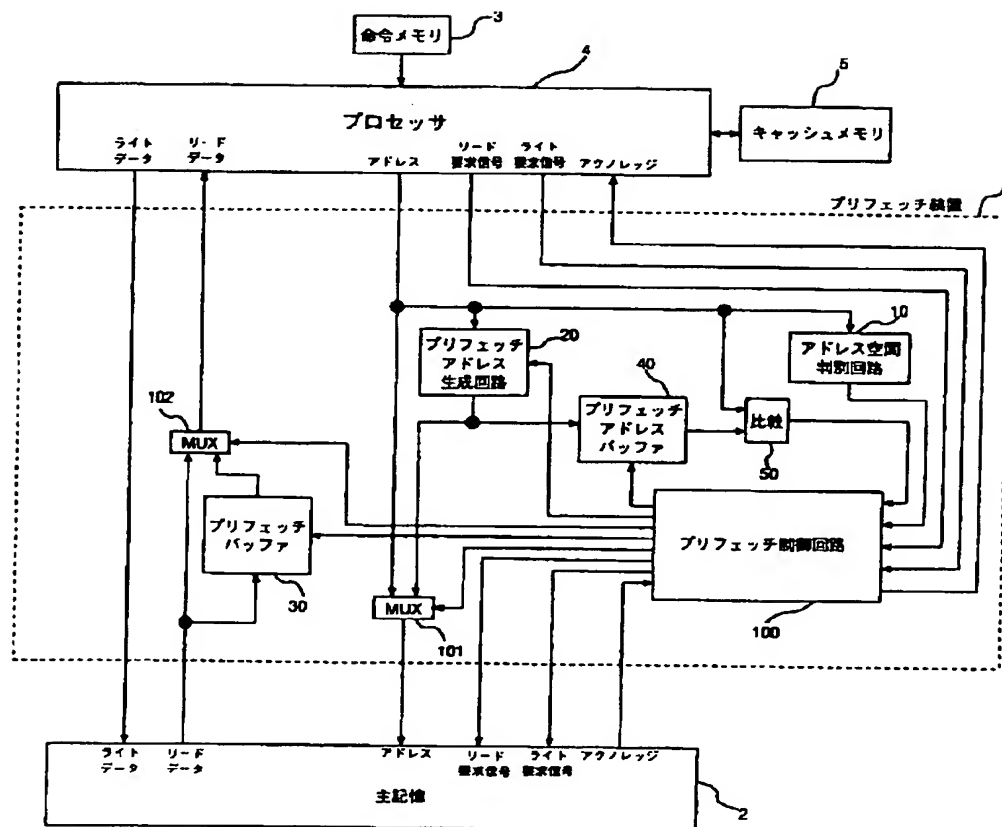
プリフェッチ制御回路100の内部状態



【図3】



【図1】



【図4】

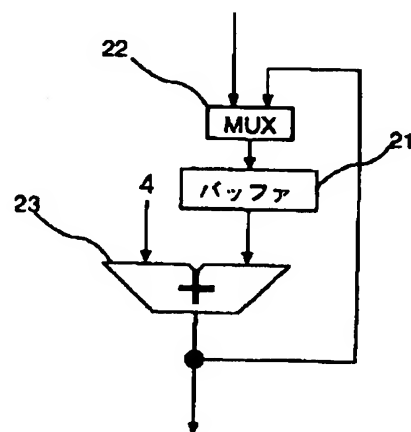
```

i0      mov    0x40000100,r0
i1      mov    0x80000100,r1
i2      mov    256,r2
i3  loop: ld    (r0),r3
i4      ld    (r1),r4
i5      mac   r3,r4
i6      add   4,r0
i7      add   4,r1
i8      add   -1,r2
i9      cmp   0,r2
i10     bgt   loop

```

【図6】

プリフェッチアドレス生成回路



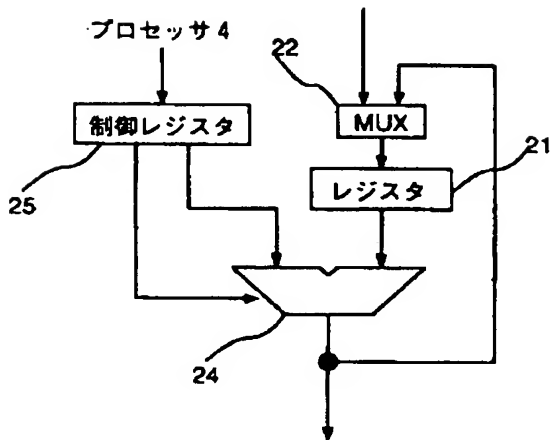


The diagram illustrates the timing of various signals over 20 clock cycles, divided into two 10-cycle loops. The signals and their states are as follows:

- Processor Command (プロセッサの実行命令):** C0 (10), C1 (11), C2 (12), C3 (13), C4 (14), C5 (15), C6 (16), C7 (17), C8 (18), C9 (19), C10 (20), C11 (21), C12 (22), C13 (23), C14 (24), C15 (25), C16 (26), C17 (27), C18 (28), C19 (29), C20 (30).
- Processor Address (プロセッサのアドレス):** 0x40000100 (C0-C1), 0x40000104 (C2-C3), 0x40000108 (C4-C5), 0x4000010C (C6-C7), 0x40000110 (C8-C9), 0x40000114 (C10-C11), 0x40000118 (C12-C13), 0x4000011C (C14-C15), 0x40000120 (C16-C17), 0x40000124 (C18-C19), 0x40000128 (C20-C21).
- Read Request (リード要求):** Active at C0, C2, C4, C6, C8, C10, C12, C14, C16, C18, C20.
- Acknowledge (アクノレッジ):** Active at C1, C3, C5, C7, C9, C11, C13, C15, C17, C19.
- Prefetch Request (プリフェッチ要求):** Active at C10, C12, C14, C16, C18, C20.
- Prefetch Enable (プリフェッチ使能):** Active at C14, C16, C18, C20.
- Prefetch Status (プリフェッチ状態):** Shows the state of the prefetch buffer, with values like 0x40000100, 0x40000104, 0x40000108, 0x40000110, 0x40000114, 0x40000118, 0x4000011C, 0x40000120, 0x40000124, 0x40000128, 0x4000012C, 0x40000130, 0x40000134, 0x40000138, 0x4000013C, 0x40000140, 0x40000144, 0x40000148, 0x4000014C, 0x40000150, 0x40000154, 0x40000158, 0x4000015C, 0x40000160, 0x40000164, 0x40000168, 0x4000016C, 0x40000170, 0x40000174, 0x40000178, 0x4000017C, 0x40000180, 0x40000184, 0x40000188, 0x4000018C, 0x40000190, 0x40000194, 0x40000198, 0x4000019C, 0x400001A0, 0x400001A4, 0x400001A8, 0x400001AC, 0x400001B0, 0x400001B4, 0x400001B8, 0x400001BC, 0x400001C0, 0x400001C4, 0x400001C8, 0x400001CC, 0x400001D0, 0x400001D4, 0x400001D8, 0x400001DC, 0x400001E0, 0x400001E4, 0x400001E8, 0x400001EC, 0x400001F0, 0x400001F4, 0x400001F8, 0x400001FC, 0x40000200, 0x40000204, 0x40000208, 0x4000020C, 0x40000210, 0x40000214, 0x40000218, 0x4000021C, 0x40000220, 0x40000224, 0x40000228, 0x4000022C, 0x40000230, 0x40000234, 0x40000238, 0x4000023C, 0x40000240, 0x40000244, 0x40000248, 0x4000024C, 0x40000250, 0x40000254, 0x40000258, 0x4000025C, 0x40000260, 0x40000264, 0x40000268, 0x4000026C, 0x40000270, 0x40000274, 0x40000278, 0x4000027C, 0x40000280, 0x40000284, 0x40000288, 0x4000028C, 0x40000290, 0x40000294, 0x40000298, 0x4000029C, 0x400002A0, 0x400002A4, 0x400002A8, 0x400002AC, 0x400002B0, 0x400002B4, 0x400002B8, 0x400002BC, 0x400002C0, 0x400002C4, 0x400002C8, 0x400002CC, 0x400002D0, 0x400002D4, 0x400002D8, 0x400002DC, 0x400002E0, 0x400002E4, 0x400002E8, 0x400002EC, 0x400002F0, 0x400002F4, 0x400002F8, 0x400002FC, 0x40000300, 0x40000304, 0x40000308, 0x4000030C, 0x40000310, 0x40000314, 0x40000318, 0x4000031C, 0x40000320, 0x40000324, 0x40000328, 0x4000032C, 0x40000330, 0x40000334, 0x40000338, 0x4000033C, 0x40000340, 0x40000344, 0x40000348, 0x4000034C, 0x40000350, 0x40000354, 0x40000358, 0x4000035C, 0x40000360, 0x40000364, 0x40000368, 0x4000036C, 0x40000370, 0x40000374, 0x40000378, 0x4000037C, 0x40000380, 0x40000384, 0x40000388, 0x4000038C, 0x40000390, 0x40000394, 0x40000398, 0x4000039C, 0x400003A0, 0x400003A4, 0x400003A8, 0x400003AC, 0x400003B0, 0x400003B4, 0x400003B8, 0x400003BC, 0x400003C0, 0x400003C4, 0x400003C8, 0x400003CC, 0x400003D0, 0x400003D4, 0x400003D8, 0x400003DC, 0x400003E0, 0x400003E4, 0x400003E8, 0x400003EC, 0x400003F0, 0x400003F4, 0x400003F8, 0x400003FC, 0x40000400, 0x40000404, 0x40000408, 0x4000040C, 0x40000410, 0x40000414, 0x40000418, 0x4000041C, 0x40000420, 0x40000424, 0x40000428, 0x4000042C, 0x40000430, 0x40000434, 0x40000438, 0x4000043C, 0x40000440, 0x40000444, 0x40000448, 0x4000044C, 0x40000450, 0x40000454, 0x40000458, 0x4000045C, 0x40000460, 0x40000464, 0x40000468, 0x4000046C, 0x40000470, 0x40000474, 0x40000478, 0x4000047C, 0x40000480, 0x40000484, 0x40000488, 0x4000048C, 0x40000490, 0x40000494, 0x40000498, 0x4000049C, 0x400004A0, 0x400004A4, 0x400004A8, 0x400004AC, 0x400004B0, 0x400004B4, 0x400004B8, 0x400004BC, 0x400004C0, 0x400004C4, 0x400004C8, 0x400004CC, 0x400004D0, 0x400004D4, 0x400004D8, 0x400004DC, 0x400004E0, 0x400004E4, 0x400004E8, 0x400004EC, 0x400004F0, 0x400004F4, 0x400004F8, 0x400004FC, 0x40000500, 0x40000504, 0x40000508, 0x4000050C, 0x40000510, 0x40000514, 0x40000518, 0x4000051C, 0x40000520, 0x40000524, 0x40000528, 0x4000052C, 0x40000530, 0x40000534, 0x40000538, 0x4000053C, 0x40000540, 0x40000544, 0x40000548, 0x4000054C, 0x40000550, 0x40000554, 0x40000558, 0x4000055C, 0x40000560, 0x40000564, 0x40000568, 0x4000056C, 0x40000570,

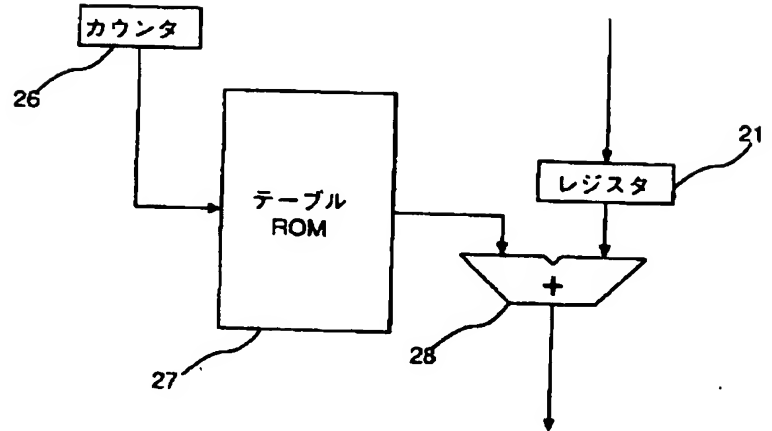
【図7】

プリフェッチアドレス生成回路

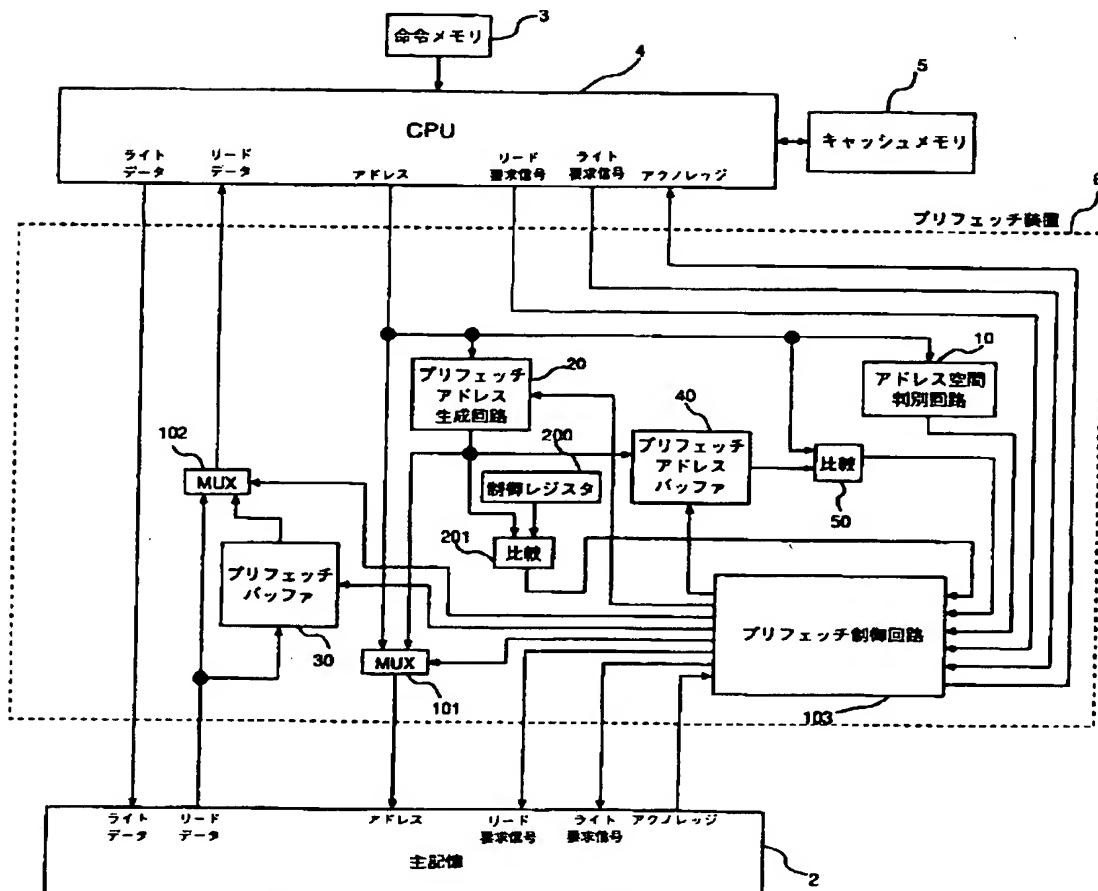


【図8】

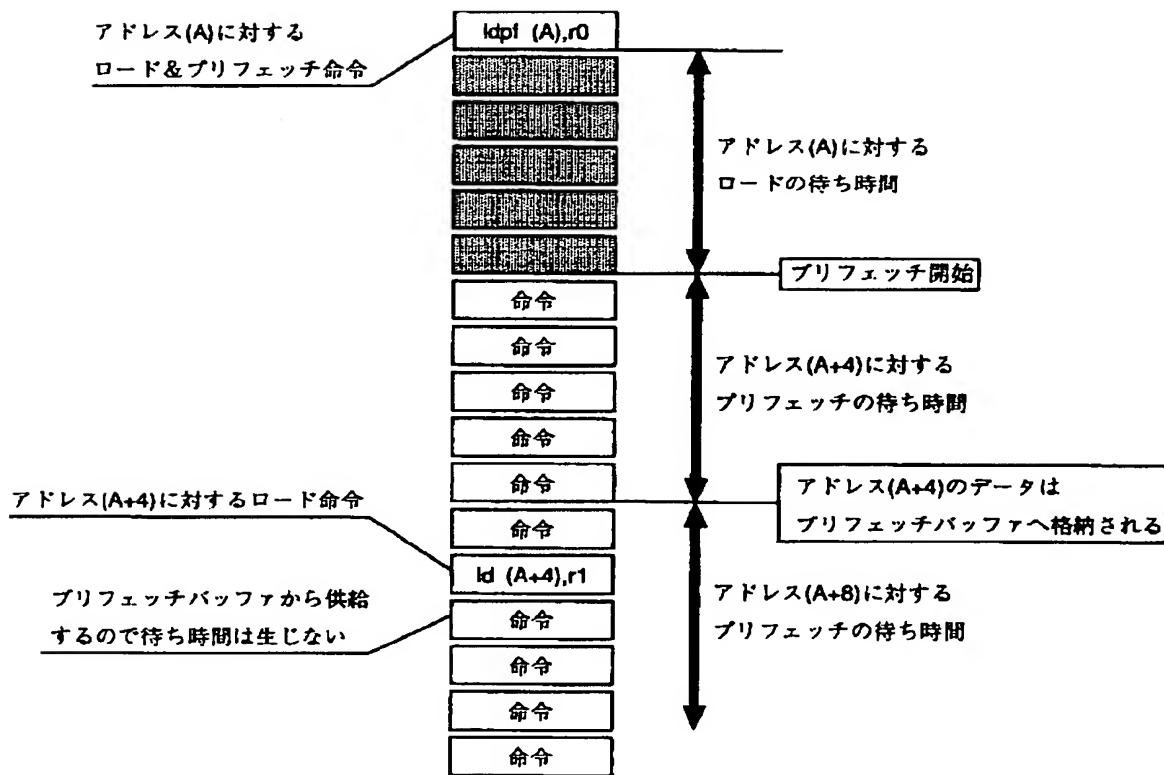
プリフェッチアドレス生成回路



【図9】



【図10】



【図11】

